DUMPSBOSS.

PeopleCert DevSecOps Exam

PEOPLECERT DevSecOps

Version Demo

Total Demo Questions: 6

Total Premium Questions: 67

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Realizing DevSecOps Outcomes	4
Topic 2, Defining the Cyber Threat Landscape	18
Topic 3, Building a Responsive DevSecOps Model	5
Topic 4, Integrating DevSecOps Stakeholders	3
Topic 5, Establishing DevSecOps Best Practices	34
Topic 6, Best Practices to Get Started	3
Total	67

QUESTION NO: 1

In shift-left thinking software Dogs and errors should IDEALLY be detected during which phase of testing?

A. During UAT tests

B. During staging tests

C. During unit tests

D. During system tests

ANSWER: C

Explanation:

During unit tests is correct because shift-left thinking moves quality, security, and defect detection as far upstream in the software delivery lifecycle as practical. Unit tests are executed against small, isolated units of code—such as functions, methods, or classes—while developers are actively implementing changes. Detecting bugs and errors at this point provides the quickest feedback, makes the underlying cause easier to isolate, and enables correction before the code is integrated with other components or promoted through later environments. This early feedback is central to DevSecOps: automated unit testing is commonly incorporated into continuous integration pipelines so that each code change can be validated consistently and rapidly. Early detection also reduces the cost and effort of remediation, because fewer downstream dependencies, test data sets, deployments, and teams are involved. IBM describes shift-left testing as testing earlier in the development process to find and fix defects sooner: IBM: Shift-left testing. The AWS DevOps guidance likewise emphasizes automated tests and fast feedback within continuous integration practices: AWS Well-Architected DevOps Guidance.

QUESTION NO: 2

Which of following BESTdescribes the types of identity-confirming credentials in four-factor authentication?

1. Recognition

2. Ownership

3. Knowledge

4. inherence

A. 1 and 2

B. 3 and 3

C. 3 and 4

D. 1 and 4

ANSWER: C

Explanation:

3 and 4 is correct because knowledge and inherence are established authentication-factor categories used to confirm a claimant's identity. A knowledge factor is evidence known by the user, such as a password, passphrase, or PIN. Its value lies in the user being able to provide information that should not be available to an unauthorized person. An inherence factor is evidence intrinsic to the user, commonly verified through a biometric characteristic such as a fingerprint, facial feature, or iris pattern. These factors are independent forms of identity evidence, which is the central principle of multi-factor and four-factor authentication: authentication confidence is strengthened by combining evidence from distinct categories rather than relying on one credential alone. NIST's digital identity guidance describes authentication using memorized secrets and biometric characteristics, while OWASP identifies knowledge and inherence among the standard factor types used in multi-factor authentication. See NIST SP 800-63B and the OWASP Multifactor Authentication Cheat Sheet.

QUESTION NO: 3

Which of the following BEST describes automated security testing?

A. Ensures that automated orchestration and provisioning software covers the scope of the application stack

B. Ensures that continuous delivery pipelines integrate testing suites and capabilities into their toolchains

C. Ensures that infrastructure and networks are software defined to enable rapid and reliable deployments

D. Ensures that applications are developed to deliver the expected results and reveal any programming errors early

ANSWER: B

Explanation:

Ensures that continuous delivery pipelines integrate testing suites and capabilities into their toolchains is correct because automated security testing embeds repeatable security checks directly into the delivery workflow. Rather than relying on a separate, manual security activity near release, teams configure pipeline stages to invoke appropriate security-testing capabilities whenever code, dependencies, infrastructure definitions, or build artifacts change. Depending on the delivery stage and technology, these capabilities can include static application security testing, software-composition analysis, secret detection, dynamic testing, container-image scanning, and checks of infrastructure-as-code.

This implementation supports the DevSecOps objective of making security feedback rapid, consistent, and actionable for the people making changes. Pipeline integration also provides a reliable mechanism for applying agreed quality and security gates, recording results, and preventing promotion of artifacts that fail required controls. The NIST Secure Software Development Framework identifies the use of automated tools and processes to find vulnerabilities throughout software development and to support secure releases. OWASP likewise describes integrating automated security testing into CI/CD as a practical way to shift security testing earlier and run it continuously. See NIST SP 800-218: Secure Software Development Framework and OWASP DevSecOps Guideline.

QUESTION NO: 4

When of the following BEST describes the type of data that requires both thesender and receiver to have encrypt/decrypt capacities?

A. Data in database

B. Data in local files

C. Data in email message

D. Data in memory card

ANSWER: C

Explanation:

Data in email message is correct because protecting an email's contents with end-to-end message encryption depends on cryptographic capability at both ends of the exchange. The sender uses the recipient's public key (or an agreed symmetric key) to encrypt the message so that its content is unreadable to unauthorized parties while it travels through email infrastructure. The intended receiver must then have the corresponding private key, passphrase, and compatible software or service to decrypt and read it. This is a core property of secure email approaches such as OpenPGP and S/MIME: encryption is applied by the sender and decryption is performed by the recipient. In a DevSecOps context, this protects sensitive information exchanged through email—including credentials, incident details, configuration material, and customer data—by ensuring confidentiality beyond the security of individual mail-server connections. The U.S. National Institute of Standards and Technology describes encryption as transforming data so that only authorized entities can recover the original information, while OpenPGP defines interoperable mechanisms for encrypted email messages. See NIST's definition of encryption and the OpenPGP Message Format specification.

QUESTION NO: 5

Which of the followingBEST describes how containers and image layers are related?

A. Layers of a container are dependent on the layer immediately above it

B. A layer within a container is designed withinmicroservicesarchitecture

C. Layers are immutable files that represent a snapshot of a container.

D. A layer consists of multiple containers withsimilarmicroservices architecture

ANSWER: C

Explanation:

“Layers are immutable files that represent a snapshot of a container.” is the best description. Container images are assembled from a sequence of filesystem layers. Each layer captures the filesystem changes introduced by an image-build instruction, such as adding files, installing packages, or setting up an application. Once created, those image layers are immutable, which enables them to be cached, reused, and shared efficiently between images and containers that use the same underlying content.

When an image is used to start a container, the runtime combines its read-only image layers into a unified filesystem view and adds a thin writable container layer on top. Changes made while the container runs are recorded in that writable layer rather than changing the original image layers. This layering model supports repeatable deployments: the image remains an unchanged, versioned artifact, while each container has its own isolated runtime changes. Docker’s documentation describes images as a stack of read-only layers and containers as adding a writable layer; its image-layer documentation also explains how layers record filesystem changes and are reused across builds. See Docker: Understanding image layers and Docker: Storage drivers.

QUESTION NO: 6

Which of the following BEST describes an example of an insider threat?

A. Non-malicious attackers

B. Other competitors

C. The general public

D. Disgruntled employees

ANSWER: D

Explanation:

Disgruntled employees is correct because an insider threat originates from an individual who has authorized access to an organization’s systems, information, facilities, or other assets and then misuses that access, intentionally or unintentionally. Employees are common insider-threat actors because their legitimate role can provide familiarity with business processes, access to sensitive data, and the ability to bypass or undermine controls that are designed primarily to protect against external actors. A disgruntled employee may deliberately misuse these privileges in response to dissatisfaction, perceived unfair treatment, or an intent to harm the organization—for example, by stealing confidential information, sabotaging a service, or disclosing credentials. In DevSecOps, reducing this risk requires security practices throughout the delivery lifecycle, including least-privilege access, separation of duties, audit logging, prompt removal of access when roles change, and monitoring for unusual activity. The U.S. Cybersecurity and Infrastructure Security Agency describes insider threats as risks posed by individuals with authorized access who may wittingly or unwittingly cause harm. The U.S. National Institute of Standards and Technology also identifies malicious insiders as a key security concern requiring governance and technical safeguards. See CISA Insider Threat Mitigation and NIST SP 800-53 Rev. 5.